# Enkripsi Pesan pada E-Mail dengan Menggunakan *Chaos Theory*

Arifin Luthfi P - 13508050
*Program Studi Teknik Informatika*
*Sekolah Teknik Elektro dan Informatika*
*Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia*
*if18050@students.if.itb.ac.id*

***E-Mail merupakan sebuah metode untuk saling bertukar pesan digital antar individu di dunia maya. Terkadang E-Mail menjadi sangat penting untuk dijaga kerahasiaannya. E-Mail yang penting bisa saja disadap oleh orang yang tidak bertanggungjawab dan diketahui pesannya. Untuk itu, metode enkripsi pesan untuk E-Mail menjadi sangat penting.***

***Chaos Theory merupakan sebuah teori yang mengatakan bahwa perubahan sekecil apapun dapat membuat perubahan besar kedepannya. Perubahan yang terjadi kedepannya hampir tidak dapat diterka, dan mendekati nilai random. Nilai random sangatlah penting untuk enkripsi pesan karena nilai yang sulit diterka akan membuat serangan/percobaan untuk mengetahui pesan asli yang sudah dienkripsi menjadi sulit atau bahkan tidak mungkin. Chaos Theory dapat digunakan karena bagaimanapun nilai random yang ada kedepannya, dibutuhkan sebuah inisialisasi yang dapat dijadikan sebuah kunci enkripsi maupun dekripsi pesan.***

***Pada makalah pengganti UAS kali ini, akan dibuat sebuah addin untuk mail client yang dapat digunakan untuk melakukan enkripsi terhadap isi pesan yang berupa teks. Proses enkripsi maupun dekripsi pada pesan akan dilakukan dengan memanfaatkan pembangkitan bilangan acak semu berdasarkan chaos theory yang sudah dipelajari di kelas.***

***Kata Kunci : Kriptografi, Chaos Theory, Bilangan Acak, E-mail, Plainteks, Cipherteks, Random, Bilangan Acak.***

## I. PENDAHULUAN

Kriptografi merupakan sebuah metode untuk menjaga kerahasiaan pesan dari pihak yang tidak berkepentingan. Pesan dirahasiakan dengan cara mengacak nilai-nilai yang terdapat didalamnya sehingga membuat pesan tersebut tidak memiliki arti lagi. Kriptografi yang ideal adalah kriptografi yang menghasilkan pesan yang nilainya bersifat random sehingga tidak dapat dipecahkan oleh pihak yang tidak mengetahui kunci untuk memecahkannya. Banyak sekali cara yang dapat digunakan orang untuk melakukan kriptografi terhadap sebuah pesan. Namun begitu, belum ada algoritma kriptografi yang benar-benar aman.

Nilai acak, atau mungkin acak semu, yang digunakan untuk melakukan proses enkripsi dan dekripsi pada kriptografi dapat diperoleh dengan berbagai cara. Salah satu cara yang bisa digunakan adalah dengan memanfaatkan *Chaos Theory*. Nilai bilangan acak yang diperoleh dengan memanfaatkan *Chaos Theory* sensitif terhadap nilai awal yang ditentukan. Pada pengaplikasian kriptografi dengan menggunakan *Chaos Theory*, nilai awal ini dapat kita jadikan kunci sehingga kita dapat menemukan kembali bilangan acak yang dihasilkan sesuai dengan kunci tersebut.

Nilai bilangan acak yang dihasilkan dari pembangkitan menggunakan *Chaos Theory* dapat digunakan untuk menggeser nilai dari sebuah pesan. Karena nilai yang dihasilkan acak, maka pergeseran nilai yang terjadi tidaklah sama. Karakter-karakter yang sama pada plainteks tidak berubah menjadi karakter yang sama pada cipherteks. Misalkan karakter "A" pada awal plainteks digeser menjadi "B", bisa saja karakter "A" yang berikutnya ditemukan digeser menjadi "C". Ini disesuaikan dengan bilangan acak yang dihasilkan.

Tidak seperti *Vigenere Cipher* dimana pergeseran karakter yang terjadi pada proses enkripsi memiliki periode sesuai dengan kuncinya, proses enkripsi yang dilakukan dengan menggunakan *Chaos Theory* tidak memiliki periode. Ini karena bilangan acak yang dihasilkan itu sendiri tidak memiliki periode, sehingga kita tidak dapat memecahkan cipherteks yang ada dengan mencari perulangan yang terjadi.

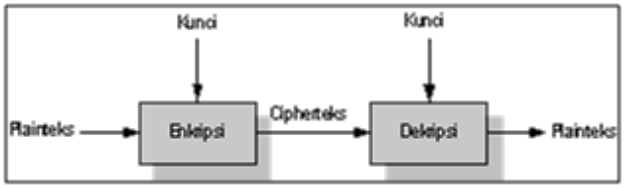
## II. TERMINOLOGI

### 2.1 Kriptografi

Kriptografi adalah ilmu dan seni untuk menjaga kerahasian pesan dengan cara menyandikannya ke dalam bentuk yang tidak dapat dimengerti lagi maknanya. Pesan dalam kriptografi dapat berupa tulisan, citra, video, dan sebagainya. Proses kriptografi dilakukan agar informasi yang terdapat didalam pesan tidak bocor kepada pihak yang tidak berkepentingan saat dilakukan pengiriman pesan.

Dalam kriptografi, pesan yang belum disandikan disebut plainteks, sedangkan pesan yang telah disandikan disebut cipherteks. Terdapat dua proses pada kriptografi yang berguna untuk menyandikan dan mengekstraksi pesan yang telah disandikan. Proses tersebut antara lain enkripsi dan dekripsi. Enkripsi adalah proses menyandikan plainteks menjadi cipherteks. Sedangkan dekripsi merupakan proses mengembalikan cipherteks

menjadi plainteks semula, agar dapat diketahui informasi yang terkandung didalamnya.




*Gambar 1 : Flow Diagram Kriptografi*

Misalkan:

*C = Chiperteks*
*P = Plainteks*

Fungsi enkripsi *E* memetakan *P* ke *C*,

*E(P) = C*

Fungsi dekripsi *D* memetakan *C* ke *P*,

*D(C) = P*

Fungsi enkripsi dan dekripsi harus memenuhi sifat:

*D(E(P)) = P*

### 2.2 Chaos Theory

*Chaos Theory* menggambarkan perilaku sistem dinamis nirlanjar yang menunjukkan fenomena *chaos.* Salah satu karakteristik dari sistem *chaos* adalah kepekaannya pada kondisi awal. Sebagai hasil dari kepekaannya terhadap kondisi awal tersebut, kelakuan sistem memperlihatkan sifat muncul acak (*random*) meskipun sistem *chaos* itu sendiri deterministik (dapat didefinisikan dengan baik dan tidak punya parameter acak).

Hanya sedikit perubahan pada kondisi awal, dapat mengubah secara drastis kelakuan sistem pada jangka panjang. Jika suatu sistem dimulai dengan kondisi awal 2, maka hasil akhir dari sistem yang sama akan jauh berbeda jika dimulai dengan 2,000001 di mana 0,000001 sangat kecil sekali dan wajar untuk diabaikan. Dengan kata lain, perbedaan yang sangat kecil di awal akan menyebabkan perubahan yang sangat besar pada waktu lainnya.

Fungsi pembangkit bilangan acak *Chaos* menggunakan iterasi dan membutuhkan nilai iterasi sebelumnya. Contoh fungsi *Chaos* :

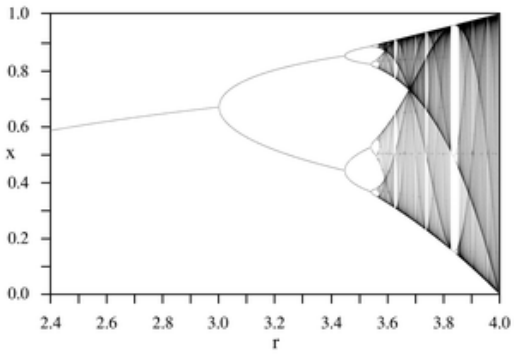
$$x_{i+1} = x_i * r\ (1 - x_i)$$

dimana

x = nilai-nilai *chaos* (0 <= x <= 1)

r = laju pertumbuhan (0 <= r <= 4)

Fungsi tersebut akan menghasilkan nilai x yang nilainya sulit diprediksi jika kita tidak mengetahui nilai x sebelumnya.




*Gambar 2: Diagram Bifurcation $x_{i+1} = x_i * r * (1 - x_i)$*

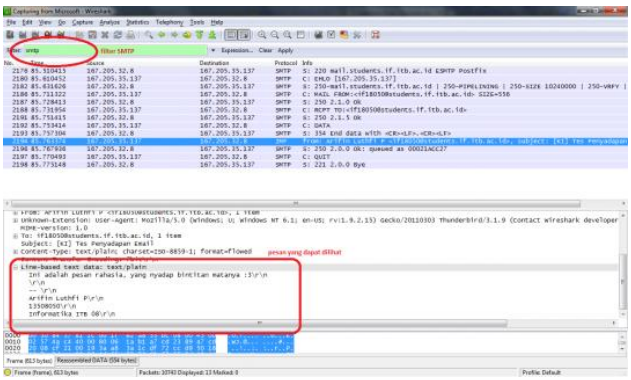
### 2.3 E-mail

E-mail atau surat adalah sarana kirim mengirim surat melalui jalur jaringan komputer (misalnya Internet). Untuk mengirim surat elektronik kita memerlukan suatu program *mail-client*. Surat elektronik yang kita kirim akan melalui beberapa poin sebelum sampai di tujuan.

Keamanan data di surat elektronik tidaklah terjamin dan selalu ada risiko terbuka untuk umum, dalam artian semua isinya dapat dibaca oleh orang lain. Hal ini disebabkan oleh karena surat elektronik itu akan melewati banyak server sebelum sampai di tujuan. Tidak tertutup kemungkinan ada orang yang menyadap surat elektronik yang dikirimkan tersebut. Surat elektronik dapat diamankan dengan melakukan teknik pengacakan (enkripsi).

## III. LINGKUP MASALAH

Isi E-mail seringkali merupakan pesan yang sangat rahasia dan hanya boleh dibaca oleh pihak yang bersangkutan. E-mail yang bersifat rahasia tidak boleh tersebar di publik sehingga publik tersebut mengetahui isi yang terdapat pada E-mail tersebut. Untuk itu, perlu ada aplikasi yang berguna untuk melakukan enkripsi terhadap E-mail agar ketika E-mail tersebut dikirimkan, publik tidak mengetahui pesan sebenarnya yang ada didalamnya walaupun E-mail tersebut bocor ke publik, terutama saat proses pengiriman.

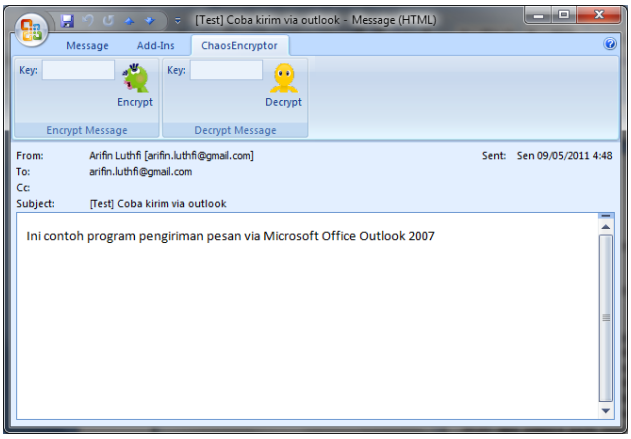


*Gambar 3: Pengiriman E-mail yang tidak aman, dapat dilakukan sniffing dengan menggunakan aplikasi Wireshark pada jaringan*

### 3.1 Add-in Enkripsi Pesan pada Mail Client

Sudah banyak aplikasi *mail client* yang dipakai oleh pengguna E-mail. Contohnya *Mozilla Thunderbird*, *Microsoft Office Outlook*, dan sebagainya. Akan sangat tepat apabila ada penambahan fitur pada aplikasi tersebut yang dapat mengenkripsi pesan E-mail yang akan dikirim dan mendekripsinya pada saat diterima. Untuk itu, pada kesempatan ini saya membuat sebuah *addin* pada *mail client Microsoft Office Outlook 2007* yang dapat digunakan untuk melakukan enkripsi maupun enkripsi pada isi pesan. *Addin* ini saya beri nama **ChaosEncryptor**.

Dibawah ini merupakan contoh tampilan dari *Microsoft Office Outlook 2007* yang telah diberi tambahan *addin* ChaosEncryptor**.** Menu enkripsi dan dekripsi terdapat pada menu ChaosEncryptor yang ada di kanan atas.




*Gambar 4: Antarmuka AddIn ChaosEncryptor*

## IV. IMPLEMENTASI

### 4.1 Metode Enkripsi Pesan

Proses enkripsi serta dekripsi pada pesan E-mail menggunakan proses pergeseran karakter alfabet berdasarkan nilai yang dihasilkan dari fungsi random dengan *Chaos Theory*. Nilai random yang dicari memiliki rentang antara 0-26, sesuai dengan jumlah alfabet. Untuk proses enkripsi, karakter digeser ke kanan. Sedangkan untuk proses dekripsi, karakter digeser ke kiri. Kita telaah contoh di bawah ini :

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
^
```

Misal karakter yang ingin dienkripsi ialah karakter ‘A’, dan nilai random yang didapat adalah 5. Maka, pada cipherteks, karakter tersebut akan diubah menjadi karakter ‘F’ (digeser 5 karakter ke kanan).

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
---->^
```

Proses enkripsi ini dilakukan untuk setiap karakter yang terdapat di dalam pesan E-mail. Sedangkan untuk proses dekripsi berlaku sebaliknya, yang semula karakter ‘F’ jika mendapat nilai random 5, maka karakter yang telah didekripsi menjadi ‘A’.

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
     ^

ABCDEFGHIJKLMNOPQRSTUVWXYZ
^<----
```

Jika pergeseran terjadi melewati batas alfabet, maka karakter akan dimulai dari awal seperti pada contoh dibawah ini (enkripsi karakter ‘W’ dengan nilai random 10):

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
                         ^

ABCDEFGHIJKLMNOPQRSTUVWXYZ
----->^                ----
```

Dengan demikian, hasil enkripsi maupun dekripsi dari pesan E-mail tidak akan keluar dari *range* alfabet, sehingga karakter tidak ada yang hilang.

Proses enkripsi dan dekripsi ini mengabaikan karakter lain selain huruf alfabet (seperti spasi, tanda tanya, dan sebagainya). Karakter lain tersebut akan tetap ditulis apa adanya pada saat proses enkripsi maupun dekripsi.

### 4.2 Pembangkitan Bilangan Acak

Diperlukan dua buah parameter untuk membangkitkan bilangan acak dengan *Chaos Theory*, yaitu nilai awal (x) dan laju pertumbuhan (r). Nilai awal merupakan masukkan dari user, yang kemudian menjadi kunci untuk proses enkripsi maupun dekripsi. Sedangkan nilai laju pertumbuhan sudah ditetapkan di dalam program. Nilai laju pertumbuhan yang dipilih adalah nilai 4. Angka tersebut dipilih agar nilai acak lebih cepat didapat.

Kunci yang diminta sebagai masukkan dari user bukanlah angka desimal dengan rentang 0-1, tapi bilangan 6 digit integer seperti pada PIN ATM. Hal ini dikarenakan sulitnya memasukkan nilai desimal dengan koma oleh user. Nilai masukkan dari user yang berupa 6 digit integer itu kemudian diubah oleh program menjadi bilangan real antara 0-1 yang merupakan syarat nilai awal dari pembangkitan bilangan acak. Nilai real dibangkitkan dengan cara membagi angka 1 dengan bilangan tersebut :

$$x_0 = 1/(input)$$

Namun, pembangkitan nilai awal bilangan acak seperti diatas menyebabkan perbedaan nilai awal yang sangat

kecil. Namun disinilah fungsi dari *Chaos Theory*, perubahan kecil pada awal dapat membuat perubahan besar kedepannya.

Kita lihat contoh yang dihasilkan masukan user sebagai nilai awal *Chaos Theory*. Perbedaan nilai awal yang dihasilkan hanyalah sedikit berbeda :

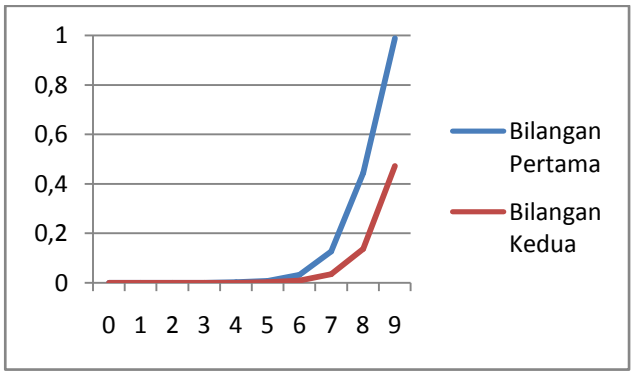
| No | Masukkan User (Kunci) | Nilai Awal ($X_0$) |
|---|---|---|
| 1 | 123456 | 8,10005184033178E-06 |
| 2 | 456789 | 2,18919457342449E-06 |

Pada tabel diatas terlihat bahwa perbedaan antara kedua nilai awal yang dihasilkan sangat kecil, hanya sekitar $6x10^{-6}$ (0,000006).

Pada awal iterasi, nilai yang dihasilkan dengan *Chaos Theory* tidaklah terlalu acak dan juga nilainya tidak jauh berbeda untuk setiap nilai awal yang diberikan. Kita lihat contoh bilangan acak yang dihasilkan dari iterasi ke-1 sampai iterasi ke-10 dengan nilai awal seperti yang ada pada tabel diatas :

| Iterasi | Bilangan Pertama | Bilangan Kedua |
|---|---|---|
| $X_0$ | 8,10005184033178E-06 | 2,18919457342449E-06 |
| $X_1$ | 3,23999449179678E-05 | 8,75675912340644E-06 |
| $X_2$ | 0,000129595580646149 | 3,50267297703044E-05 |
| $X_3$ | 0,000518315142526502 | 0,000140102011594024 |
| $X_4$ | 0,00207218596775812 | 0,000560329532081485 |
| $X_5$ | 0,00827156805229259 | 0,00224006225158785 |
| $X_6$ | 0,0328125968569955 | 0,00894017749078744 |
| $X_7$ | 0,126943721377983 | 0,0354410028688826 |
| $X_8$ | 0,443316051922769 | 0,136739752738122 |
| $X_9$ | 0,987147720121511 | 0,472167971036957 |

Jika kita buat dalam bentuk grafik, akan terlihat seperti gambar dibawah ini :




*Grafik 1: Persebaran Nilai dengan Iterasi 0-9*

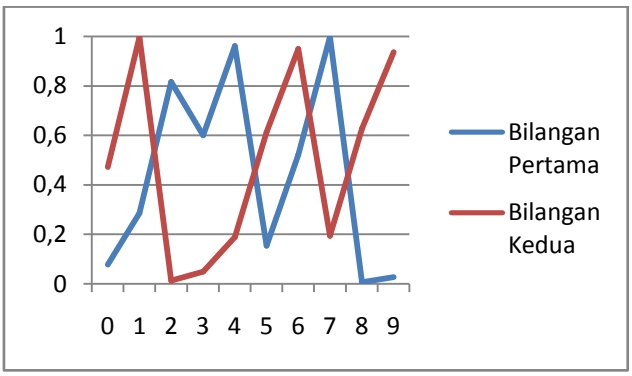
Dilihat dari data diatas, perbedaan antara kedua nilai setelah iterasi tidaklah jauh berbeda. Hal ini membuat pergeseran karakter yang digunakan pada proses enkripsi menjadi tidak terlihat berbeda untuk setiap kunci. Kita lihat tabel yang menyatakan jumlah pergeseran karakter jika nilai dalam tabel diatas dikalikan 26 :

| Iterasi | Bilangan Pertama | Bilangan Kedua |
|---|---|---|
| $X_0$ | 0 | 0 |
| $X_1$ | 0 | 0 |
| $X_2$ | 0 | 0 |
| $X_3$ | 0 | 0 |
| $X_4$ | 0 | 0 |
| $X_5$ | 0 | 0 |
| $X_6$ | 0 | 0 |
| $X_7$ | 3 | 0 |
| $X_8$ | 11 | 3 |
| $X_9$ | 25 | 12 |

Terlihat di atas untuk iterasi ke-1 sampai iterasi ke-7, kedua kondisi bilangan awal tidak menimbulkan pergeseran jika diaplikasikan untuk proses enkripsi dan dekripsi pada pesan yang menggunakan pembangkit bilangan acak. Hal ini dikarenakan pada awalnya, bilangan-bilangan tersebut nilainya sangatlah kecil, sehingga tidak menimbulkan perbedaan berarti. Nilai random yang di hasilkan dari fungsi *chaos* barulah terlihat setelah iterasi tertentu. Mari kita lihat contoh nilai yang dihasilkan pada iterasi ke 100-110 :

| Iterasi | Bilangan Pertama | Bilangan Kedua |
|---|---|---|
| $X_{100}$ | 0,0774021916488507 | 0,47190328804042 |
| $X_{101}$ | 0,285644369507221 | 0,996842299108241 |
| $X_{102}$ | 0,816206654704173 | 0,0125909192673471 |
| $X_{103}$ | 0,600053406083183 | 0,0497295520774011 |
| $X_{104}$ | 0,959957263724614 | 0,189026094910329 |
| $X_{105}$ | 0,153757262187862 | 0,61318092141312 |
| $X_{106}$ | 0,52046386604942 | 0,948760316112308 |
| $X_{107}$ | 0,998324920745246 | 0,194456714731125 |
| $X_{108}$ | 0,0066890934569788 | 0,626573203308411 |
| $X_{109}$ | 0,0265773979428104 | 0,935916896816991 |

Jika kita buat dalam bentuk grafik, akan terlihat seperti gambar dibawah ini :




*Grafik 2: Persebaran Nilai dengan Iterasi 100-109*

Pada iterasi diatas 100, dapat dilihat bahwa nilai yang dihasilkan adalah nilai yang acak. Hal ini bagus untuk melakukan proses enkripsi karena akan membuat pergeseran alfabet yang akan dilakukan tidak dapat diketahui besarnya tanpa mengetahui kuncinya. Pergeseran karakter dapat dilihat pada tabel dibawah ini :

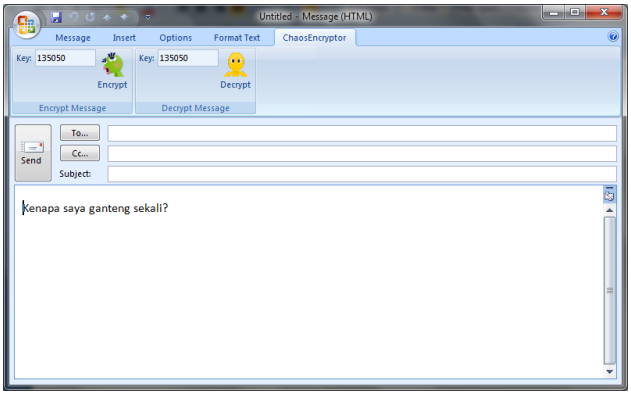
| Iterasi | Bilangan Pertama | Bilangan Kedua |
|---|---|---|
| $X_{100}$ | 2 | 12 |
| $X_{101}$ | 7 | 25 |
| $X_{102}$ | 21 | 0 |
| $X_{103}$ | 15 | 1 |
| $X_{104}$ | 24 | 4 |
| $X_{105}$ | 3 | 15 |
| $X_{106}$ | 13 | 24 |
| $X_{107}$ | 25 | 5 |
| $X_{108}$ | 0 | 16 |
| $X_{109}$ | 0 | 24 |

Dengan demikian, kita dapat mengambil kesimpulan bahwa diperlukan nilai tertentu yang harus diambil agar urutan bilangan acak dari *Chaos Theory* menjadi benar-benar acak. Pada kasus ini, angka 100 sudah cukup untuk membangkitkan bilangan yang acak.

### 4.3 Proses Enkripsi Pesan

Setelah bilangan acak dibangkitkan, pesan siap untuk dienkripsi maupun didekripsi. Proses enkripsi dan dekripsi dengan menggunakan *Chaos Theory* dimungkinkan karena bagaimanapun nilai acak yang dihasilkan, nilai itu bergantung pada nilai awal sebagai kuncinya.

Pergeseran karakter untuk proses enkripsi lebih jelas terlihat pada contoh dibawah ini :




*Gambar 5: Contoh Plainteks dalam AddIn*

Plainteks :

```
Kenapa saya ganteng sekali?
```

Hal pertama yang dilakukan adalah membangkitkan bilangan acak sebanyak panjang dari string pada plainteks. Bilangan acak tersebut dibangkitkan berdasarkan kunci masukan dari user.

Kunci masukkan user :

```
135050
```

Nilai $X_0$ yang dihasilkan dari kunci :

```
7,40466493891151E-06
```

Pembangkitan array bilangan acak dimulai dari iterasi ke 100 sejumlah panjang string (27). Laju pertumbuhan yang digunakan adalah 4 (didefinisikan dalam program) Bilangan acak yang dibangkitkan adalah :

| Iterasi | Nilai | Iterasi | Nilai |
|---|---|---|---|
| $X_{100}$ | 0,1540678936 | $X_{114}$ | 0,8623596512 |
| $X_{101}$ | 0,5213239110 | $X_{115}$ | 0,4747819326 |
| $X_{102}$ | 0,9981811632 | $X_{116}$ | 0,9974561963 |
| $X_{103}$ | 0,0072621142 | $X_{117}$ | 0,0101493309 |
| $X_{104}$ | 0,0288375038 | $X_{118}$ | 0,0401852882 |
| $X_{105}$ | 0,1120236089 | $X_{119}$ | 0,1542817234 |
| $X_{106}$ | 0,3978972798 | $X_{120}$ | 0,5219154929 |
| $X_{107}$ | 0,9583001381 | $X_{121}$ | 0,9980788446 |
| $X_{108}$ | 0,1598439335 | $X_{122}$ | 0,0076698579 |
| $X_{100}$ | 0,5371754017 | $X_{123}$ | 0,0304441249 |
| $X_{110}$ | 0,9944719580 | $X_{124}$ | 0,1180691208 |
| $X_{111}$ | 0,0219899309 | $X_{125}$ | 0,4165152142 |
| $X_{112}$ | 0,0860254956 | $X_{126}$ | 0,9721211621 |
| $X_{113}$ | 0,3145004388 | | |

Dari tabel diatas, lantas kita hitung nilai pergeseran yang akan dilakukan saat proses enkripsi dan dekripsi. Untuk mendapatkanya, kita cukup mengalikannya dengan 26 sesuai jumlah alfabet.

| Iterasi | Nilai | Iterasi | Nilai |
|---|---|---|---|
| $X_{100}$ | 4 | $X_{114}$ | 22 |
| $X_{101}$ | 13 | $X_{115}$ | 12 |
| $X_{102}$ | 25 | $X_{116}$ | 25 |
| $X_{103}$ | 0 | $X_{117}$ | 0 |
| $X_{104}$ | 0 | $X_{118}$ | 1 |
| $X_{105}$ | 2 | $X_{119}$ | 4 |
| $X_{106}$ | 10 | $X_{120}$ | 13 |
| $X_{107}$ | 24 | $X_{121}$ | 25 |
| $X_{108}$ | 4 | $X_{122}$ | 0 |
| $X_{100}$ | 13 | $X_{123}$ | 0 |
| $X_{110}$ | 25 | $X_{124}$ | 3 |
| $X_{111}$ | 0 | $X_{125}$ | 10 |
| $X_{112}$ | 2 | $X_{126}$ | 25 |
| $X_{113}$ | 8 | | |

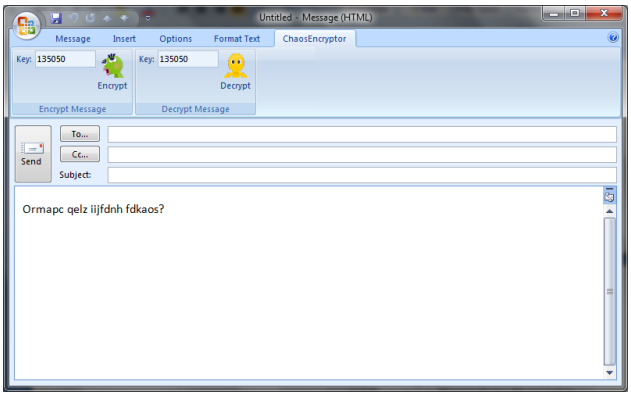
Setelah kita mendapatkan nilai pergeseran yang harus dilakukan pada proses enkripsi, kita dapat langsung mengubah tiap karakter yang ada didalam pesan.

| No | Pergeseran | Plain | Cipher |
|---|---|---|---|
| 1 | 4 | K | O |
| 2 | 13 | e | r |
| 3 | 25 | n | m |
| 4 | 0 | a | a |
| 5 | 0 | p | p |
| 6 | 2 | a | c |
| 7 | 10 | | |
| 8 | 24 | s | q |
| 9 | 4 | a | e |
| 10 | 13 | y | l |

| | | | |
|---|---|---|---|
| 11 | 25 | a | z |
| 12 | 0 | | |
| 13 | 2 | g | i |
| 14 | 8 | a | i |
| 15 | 22 | n | j |
| 16 | 12 | t | f |
| 17 | 25 | e | d |
| 18 | 0 | n | n |
| 19 | 1 | g | h |
| 20 | 4 | | |
| 21 | 13 | s | f |
| 22 | 25 | e | d |
| 23 | 0 | k | k |
| 24 | 0 | a | a |
| 25 | 3 | l | o |
| 26 | 10 | i | s |
| 27 | 25 | ? | ? |

Cipherteks :

```
Ormapc qelz iijfdnh fdkaos?
```




*Gambar 6: Contoh Cipherteks pada AddIn*

Dari proses diatas, kita dapatkan cipherteks dari plainteks yang bersangkutan. Untuk proses dekripsi, hal yang berbeda dari proses enkripsi adalah pada proses pergeseran karakter yang mana pada proses enkripsi, karakter digeser kekanan, sedangkan pada proses dekripsi karakter digeser kekiri.

## V. ANALISIS HASIL

Kita dapat menelaah contoh-contoh pesan yang sudah dienkripsi di bawah ini untuk menganalisis sejauh mana daya enkripsi pesan dengan menggunakan *Chaos Theory*.

Contoh 1 :

| |
|---|
| Kunci :<br>123321 |
| Plainteks :<br>`Ini adalah contoh plainteks yang pendek saja...` |
| Cipherteks :<br>`Eah adcuxp nnpbks rsvwmulej hyua mnljxe noib...` |

Kita coba ubah kunci sedikit saja, apakah perubahan yang terjadi berbeda jauh,

Contoh 2 :

| |
|---|
| Kunci :<br>123322 |
| Plainteks :<br>`Ini adalah contoh plainteks yang pendek saja...` |
| Cipherteks :<br>`Grw birhlg gdlxdf ekdumtgro xani ksmfmh qexz...` |

Dari contoh diatas, dapat dilihat bahwa dengan sedikit perubahan pada kunci, hasil enkripsi menjadi berbeda jauh. Hal ini disebabkan karena pembangkitan kunci dengan menggunakan *Chaos Theory* yang sangat peka terhadap kondisi awal.

Selanjutnya mari kita coba melihat pergeseran karakter yang terjadi pada huruf yang homogen pada contoh dibawah ini :

Contoh 3 :

| |
|---|
| Kunci :<br>111111 |
| Plainteks :<br>`aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa` |
| Cipherteks :<br>`aabfrwkzcjyhuqxhvnzabdnzaabepzckzdmzabgsurxjygs` |

Dapat dilihat bahwa walaupun plainteksnya homogen, cipherteks tidaklah homogen. Hal ini berarti proses enkripsi dengan menggunakan *Chaos Theory* merupakan algoritma kriptografi abjad majemuk.

Dari contoh diatas juga terlihat bahwa proses substitusi menggunakan *Chaos Theory* tidak memiliki periode seperti pada *Vigenere Cipher* dimana periodenya sesuai dengan panjang kunci yang dimasukkan.

### 5.1 Kemungkinan Serangan

Dalam kriptografi, banyak sekali metode yang bisa digunakan untuk memecahkan cipherteks oleh pihak yang tidak berkepentingan. Metode-metode yang biasa dilakukan adalah seperti metode terkaan, statistik, *exhaustive key search*, dan analisis frekuensi. Dengan enkripsi menggunakan *Chaos Theory*, kita dapat mencegah pemecahan cipheteks dengan beberapa metode yang disebutkan diatas.

1. Metode terkaan, dengan metode ini, hanya akan terlihat beberapa kata yang mungkin. Jika kita menerka dengan benar sebuah kata, maka kata yang sama di tempat lain tidak akan diketahui karena setiap karakter dienkripsi dengan pergeseran pesan yang berbeda.
2. Statistik, statistik kemunculan huruf pada plainteks tidak akan diketahui karena setiap huruf dienkripsi menjadi karakter yang berbeda.
3. *Exhaustive Key Search*, metode ini digunakan untuk menemukan kunci pada cipherteks yang dienkripsi dengan menggunakan *Vigenere Cipher*. Metode ini

tidak dapat digunakan karena tidak ada periode pada pergeseran yang dilakukan oleh enkripsi menggunakan *Chaos Theory*.

4. Analisis frekuensi, hampir sama dengan statistik, metode dengan cara menghitung kemunculan huruf pada cipherteks ini tidak berguna karena setiap karakter dienkripsi menjadi karakter yang berbeda.

## VI. SIMPULAN

E-mail dapat merupakan sebuah pesan yang sangat rahasia. E-mail dapat saja bocor kepada orang yang tidak berkepentingan pada saat proses pengiriman. Untuk itu, diperlukan proses enkripsi agar walaupun E-mail sampai ke orang yang tidak berkepentingan, pesan dalam E-mail tersebut tidak dapat diketahui.

*Chaos Theory* dapat digunakan untuk proses enkripsi pesan. Nilai acak yang dihasilkan oleh *Chaos Theory* dapat digunakan sebagai alat enkripsi karena bagaimanapun nilai acak yang dihasilkan, nilai tersebut bergantung dengan berdasarkan nilai awal yang diberikan.

Beberapa point penting dapat diambil dari proses enkripsi dengan menggunakan *Chaos Theory* antara lain adalah :

1. Proses enkripsi/dekripsi dilakukan dengan cara melakukan *shifting* (pergeseran) karakter sesuai dengan nilai yang dihasilkan *Chaos Theory*.
2. Tidak terdapat periode perulangan dalam enkripsi pesan karena bilangan acak yang dihasilkan *Chaos Theory* tidak memiliki periode.
3. Perubahan sedikit saja pada kunci, akan menyebabkan perubahan yang sangat besar pada cipheteks yang dihasilkan. Ini sesuai dengan sifat *Theory Chaos* yang digunakan dalam proses enkripsi.
4. Kemungkinan-kemungkinan serangan dengan metode kriptanalis yang sudah ada dapat diantisipasi, karena proses enkripsi dengan menggunakan *Chaos Theory* tidak memiliki periode dan pergeseran yang terjaid berbeda untuk setiap karakter.

## VII. ACKNOWLEDGMENT

Banyak terimakasih saya ucapkan kepada semua orang yang telah membantu saya dalam membuat makalah ini.

## REFERENCES

[1] Munir, Rinaldi. 2011. Bahan Kuliah IF3054 Kriptografi. Departemen Teknik Informatika, Institut Teknologi Bandung.

[2] http://www.informatika.org/~rinaldi/Kriptografi/2010-2011/Pengantar%20Kriptografi.ppt, tanggal akses 7 Mei 2011

[3] http://www.informatika.org/~rinaldi/Kriptografi/2010-2011/Pembangkit%20Bilangan%20Acak.ppt, tanggal akses 8 Mei 2011

[4] http://msdn.microsoft.com/en-us/library/bb226712%28v=office.12%29.aspx, tanggal akses 8 Mei 2011

[5] http://id.wikipedia.org/wiki/Efek_kupu-kupu, tanggal akses 8 Mei 2011

[6] http://id.wikipedia.org/wiki/Surat_elektronik, tanggal akses 8 Mei 2011

## PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 9 Mei 2011

Arifin Luthfi P
13508050